geometric representations for computer-aided design

(*from bezier splines to nurbs*)

Jonathan Balzer UCLAVISIONLAB

motivation

typesetting

ce of the two separate error terms
$\#V(g_s, g_t)$ and $\#V(g_t)$. Note tha
a SLAM algorithm: At initializat
pty, we have $V(g_t) = \emptyset$; $E_\mathrm{p} =$
uivalent to the classical BA functi
age correspondence fails, $V(g_s, g$

polygon meshes

cad

agenda

- foundations
  - vector spaces of polynomials
  - Stone-Weierstrass
- Bézier curves
  - Bernstein basis
  - de Casteljau algorithm
- B-splines
  - cardinal splines
  - nurbs
  - parametric surfaces
- demo: cad-representations in blender

foundations

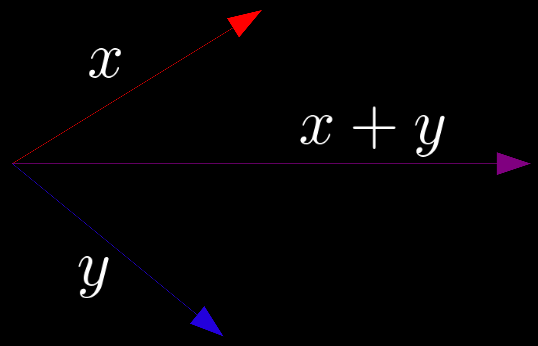
vector spaces

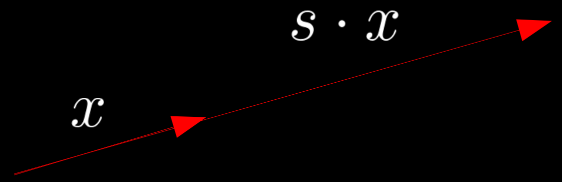
Definition 1.1. A *real vector space* is a set V such that for any two elements $x, y \in V$ and any $s \in \mathbb{R}$, the following identities hold:

(i) $x + y \in V$,
(ii) $s \cdot x \in V$.

Furthermore,

(iii) $(V, +)$ is an Abelian group,
(iv) scalar multiplication is associative and distributive,
(v) $1 \cdot x = x$.



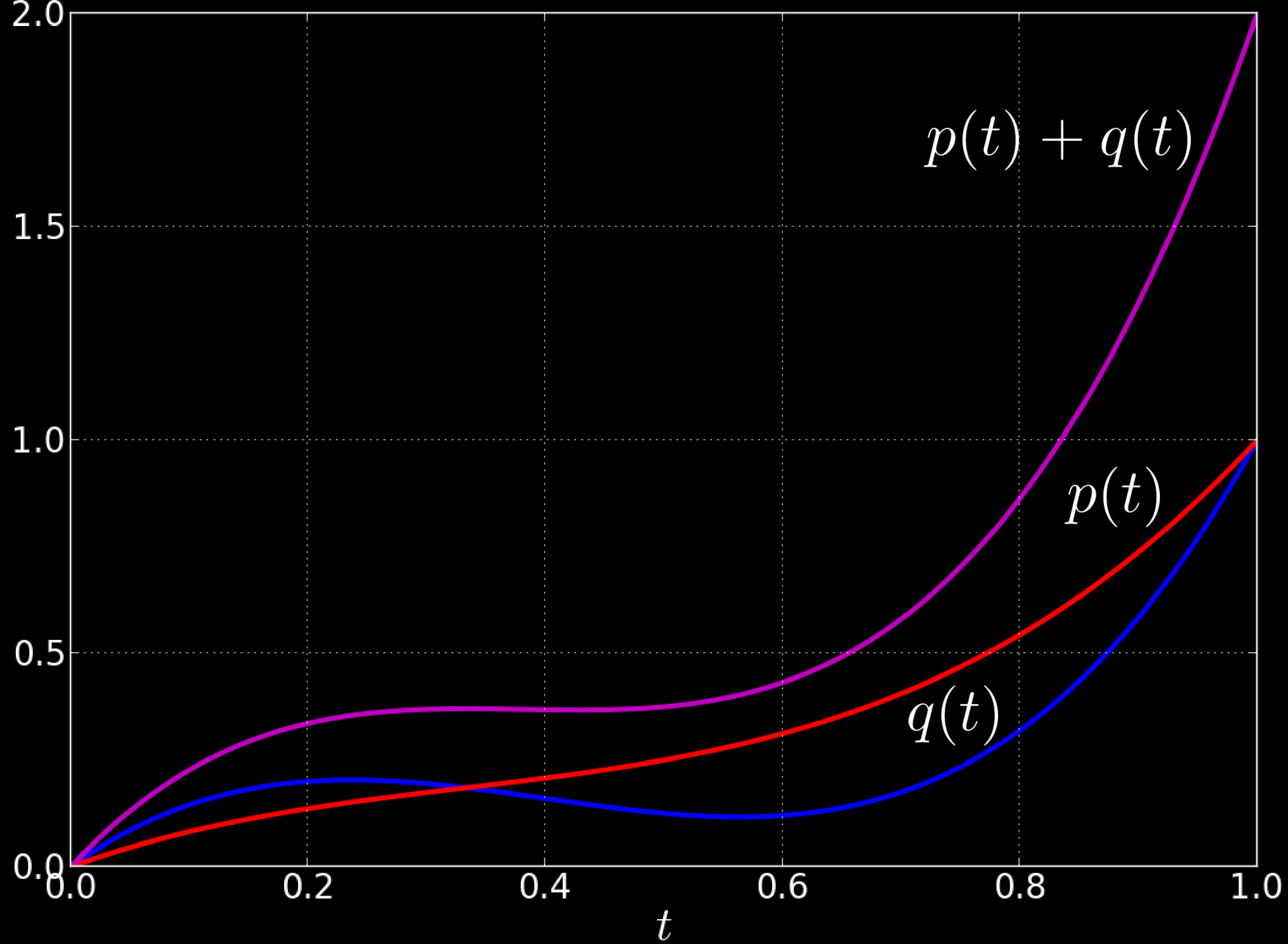


polynomials

$$p(t) = a_n t^n + a_{n-1} t^{n-1} + \ldots + a_1 t + a_0$$

- *coefficients* $a_i \in \mathbb{R}$
- *degree* $n \in \mathbb{N}$
- *order* $o = n + 1$ (number of coefficients)
- the vector space $P_n([0, 1])$

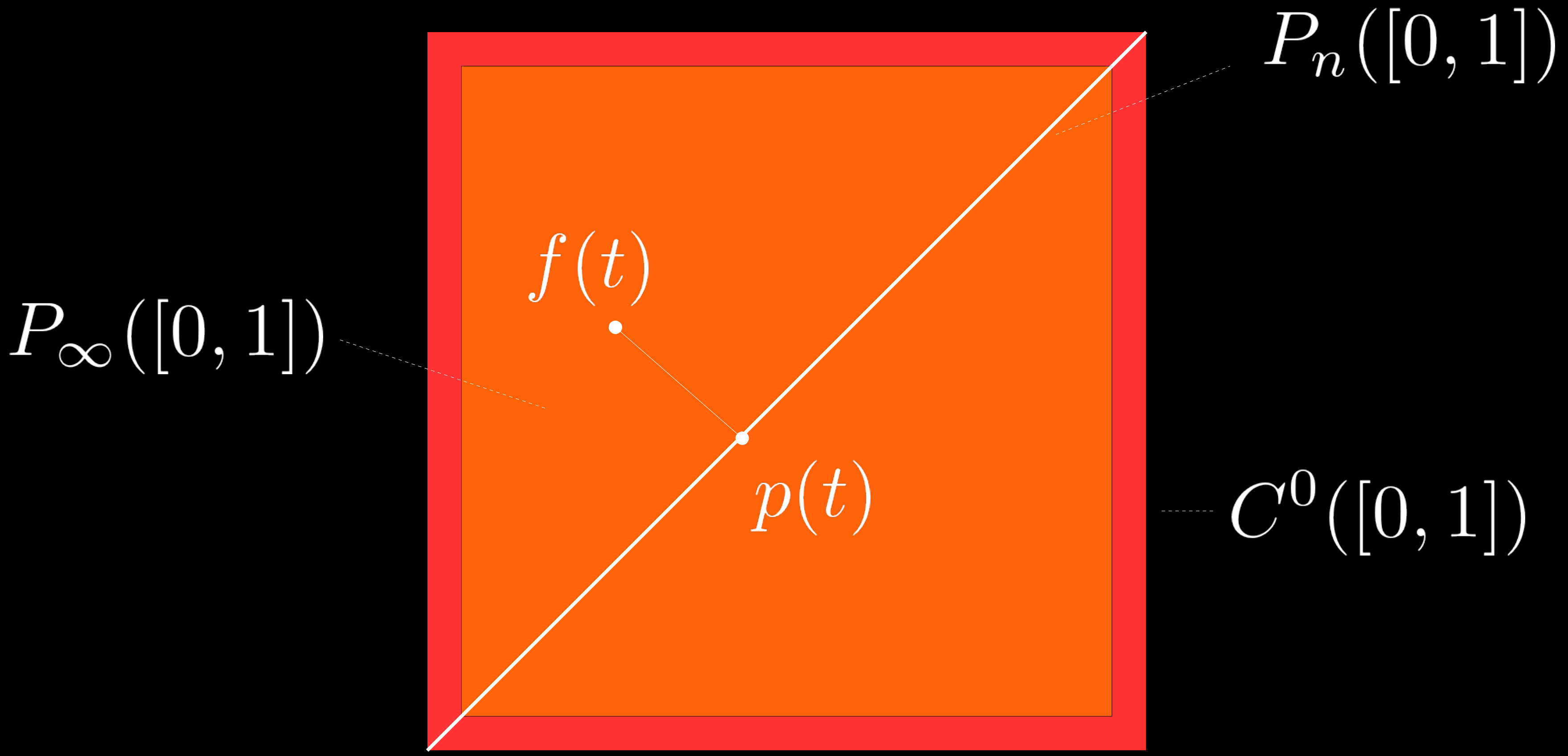
additivity



function approximation

Theorem 1.1 (Stone-Weierstrass). *Suppose* f *is a continuous function defined on the interval* $[0, 1]$. *For every* $\varepsilon > 0$, *there exists a polynomial* p *over* $[0, 1]$ *such that for all* $t \in [0, 1]$, *we have* $|f(t) - p(t)| < \varepsilon$.

function approximation



Bézier curves

Pierre Bézier

- 9/1/1910 – 11/15/1989
- MSc Mechanichal Engineering, MSc Electrical Engineering
- PhD Mathematics
- 42 year tenure at Renault

renault r4

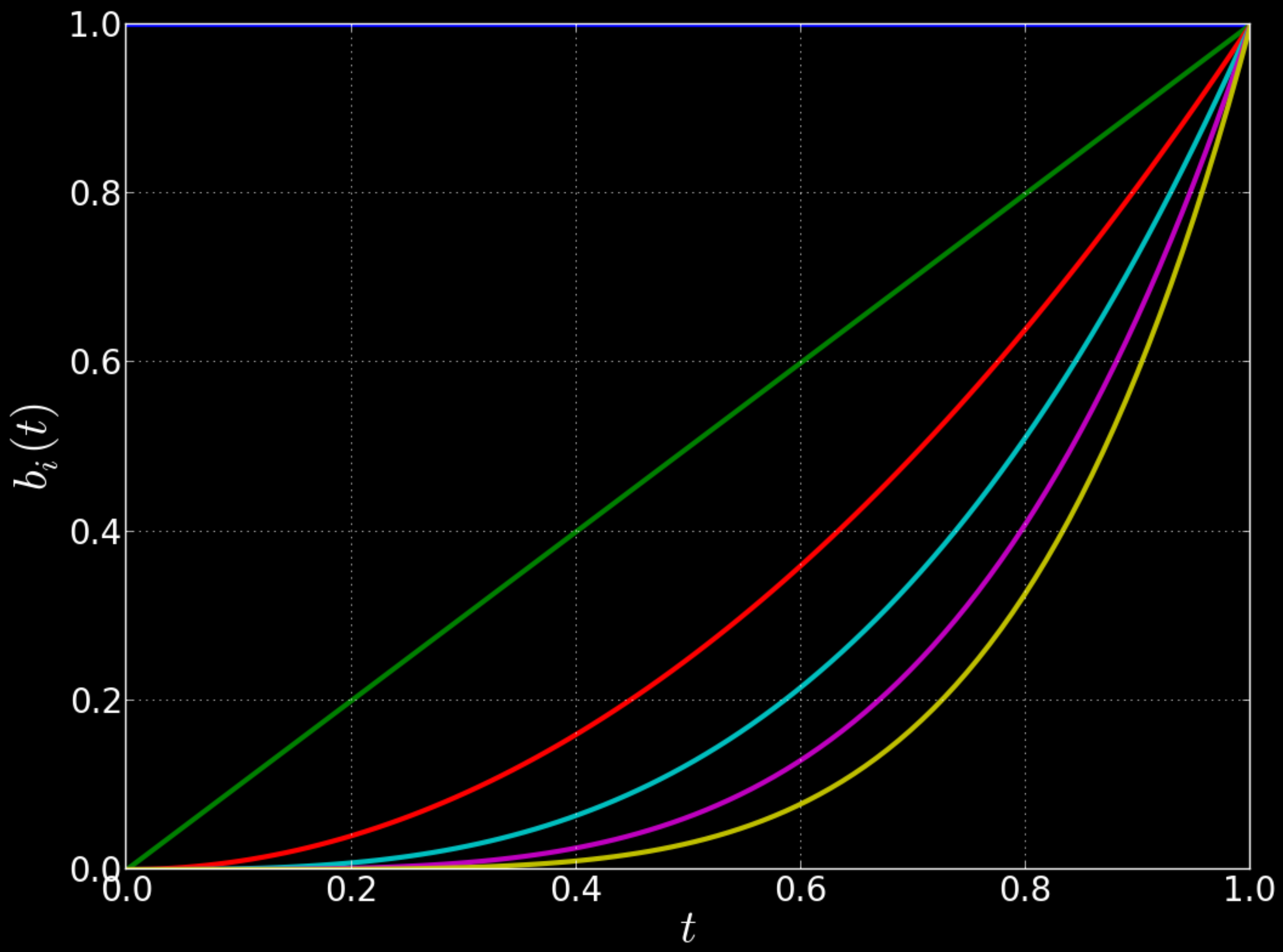
basis

- a set $\{b_1, \ldots, b_n\} \subset V$ s.t. any $x \in V$ can be written as

$$x = \sum_{i=1}^{n} c_i b_i, \quad c_i \in \mathbb{R}$$

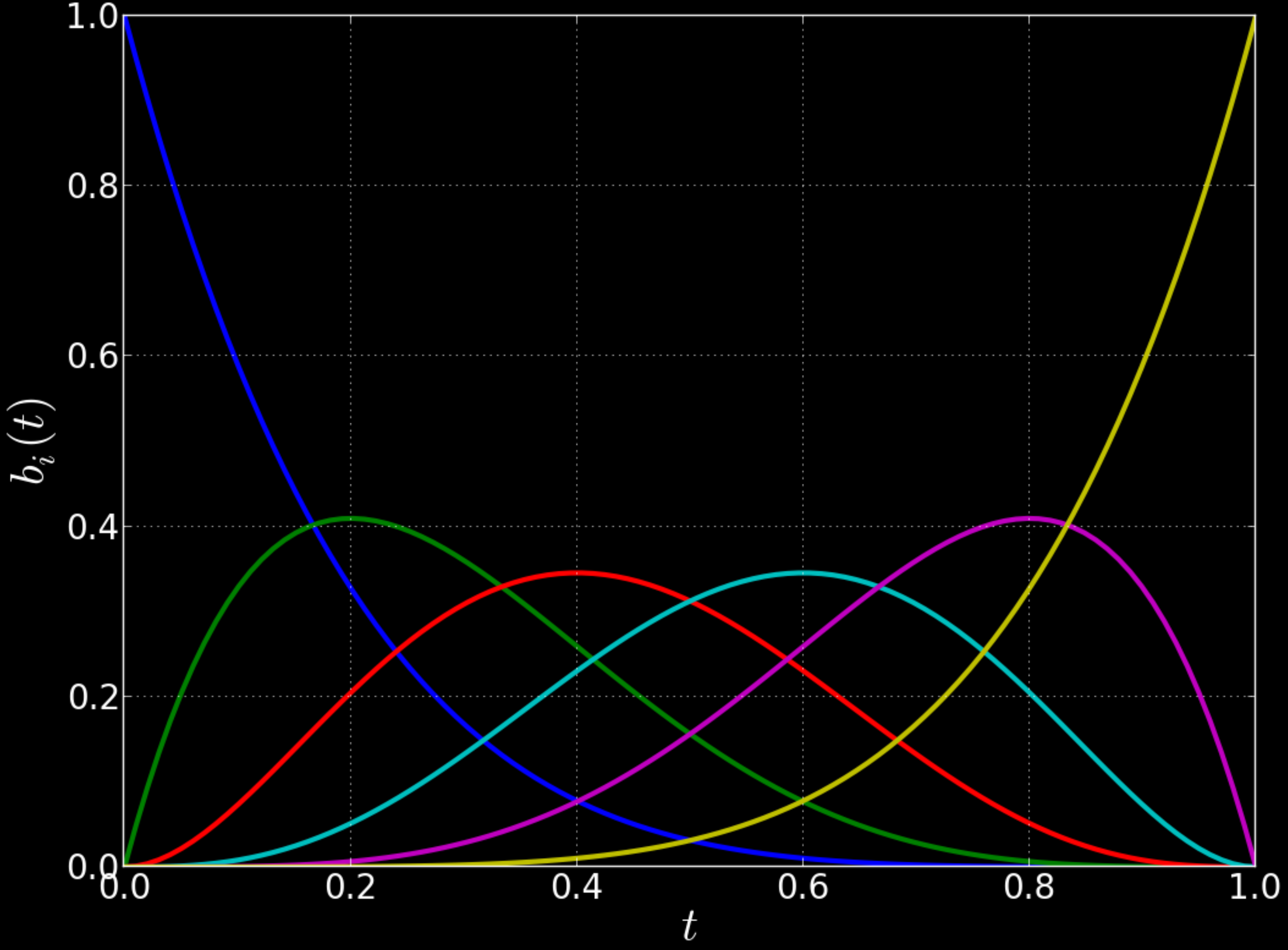
- what about $P_n([0,1])$?

monomial basis



$$b_i(t) = t^i$$

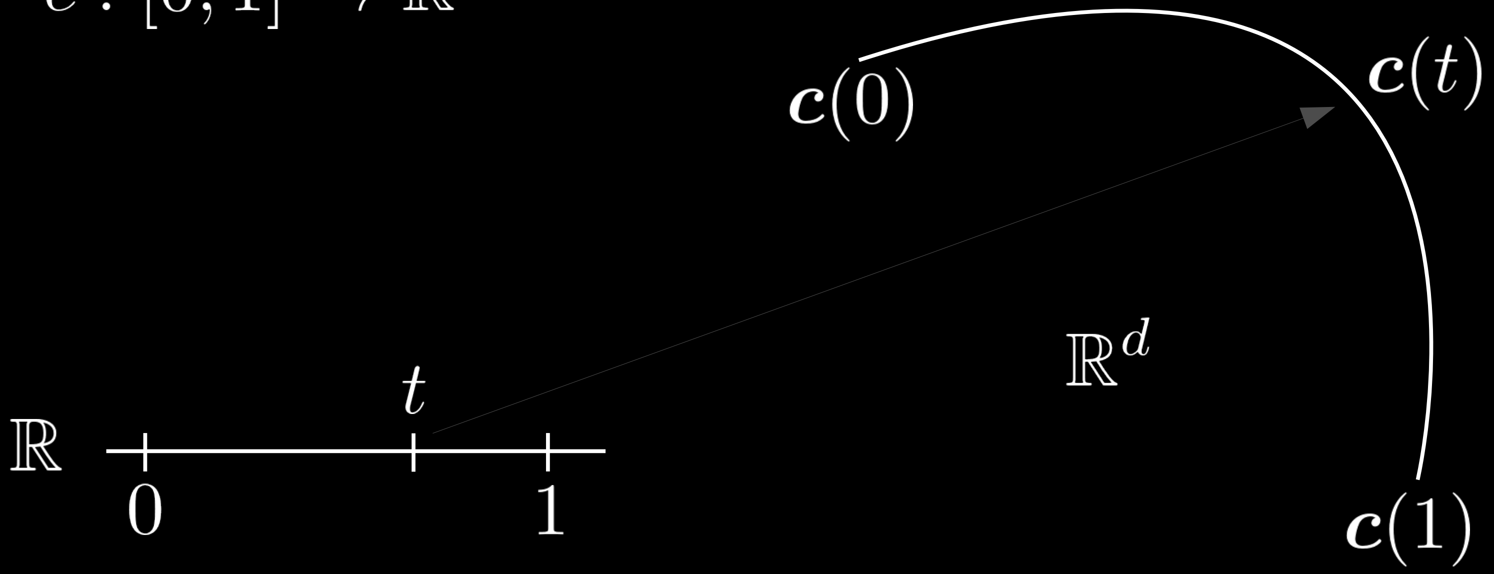
Bernstein basis



$$b_i(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

parametric curves

- map $c : [0, 1] \to \mathbb{R}^d$



- Bézier representation:

$$\boldsymbol{c}(t) = \sum_{i=1}^{n} \boldsymbol{c}_i b_i(t), \quad \boldsymbol{c}_i \in \mathbb{R}^d$$

control points

Paul de Casteljau

- born 11/19/1930
- french physicist and mathematician
- 34 years at Citroën

citroën ds

de Casteljau algorithm

- recursion: $c_i^k := c_i^{k-1}(1 - t_0) + c_{i+1}^{k-1} t_0$

$$c_0 = c_0^0$$

$$c_0^1$$

$$c_1 = c_1^0$$

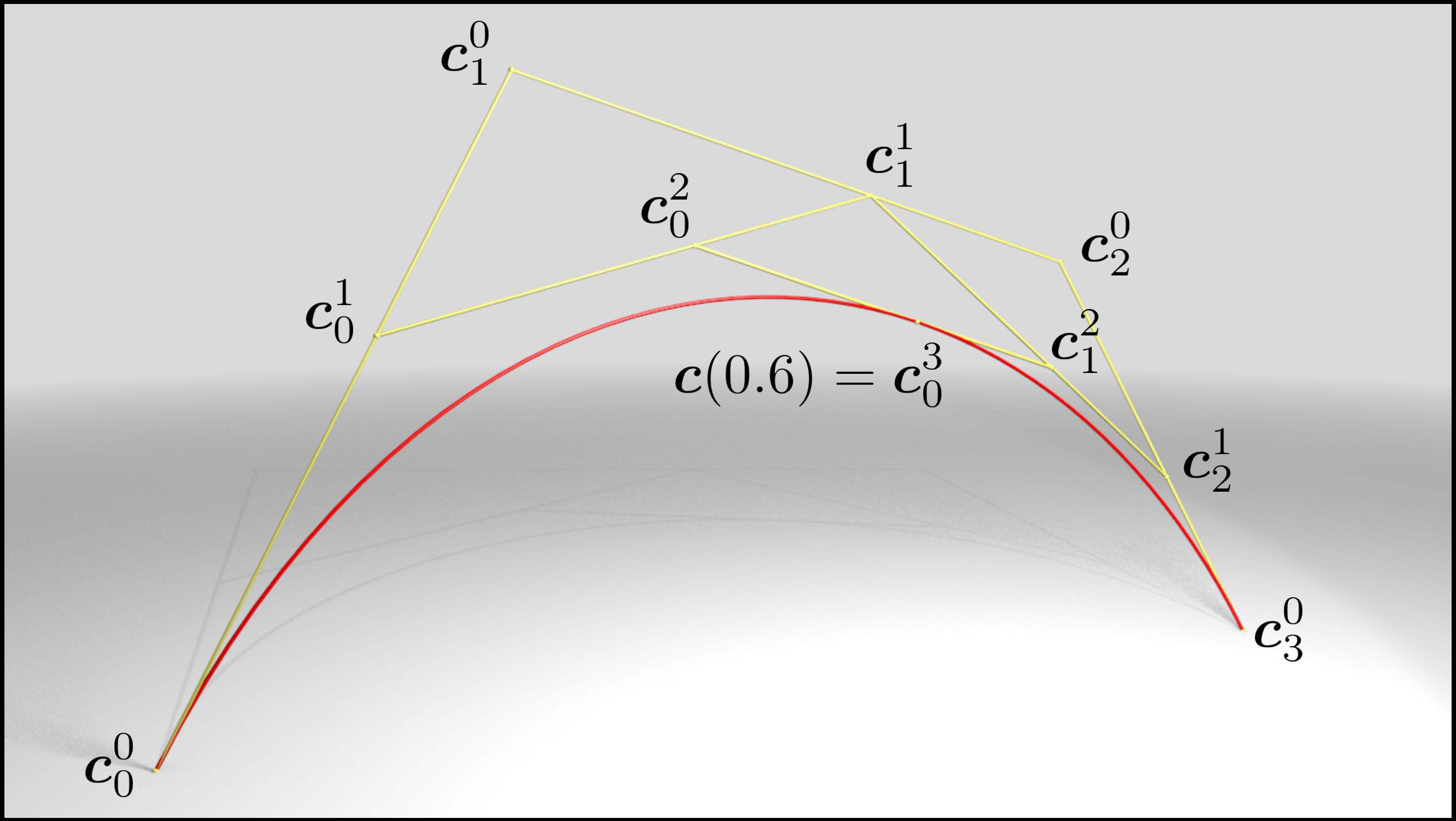
$$\vdots \qquad \vdots \qquad \ddots \qquad \boxed{c_0^n} \rightarrow p(t_0) = c_0^n$$

$$c_{n-1} = c_{n-1}^0$$

$$c_{n-1}^1$$

$$c_n = c_n^0$$

visualization $t_0 = 0.6$

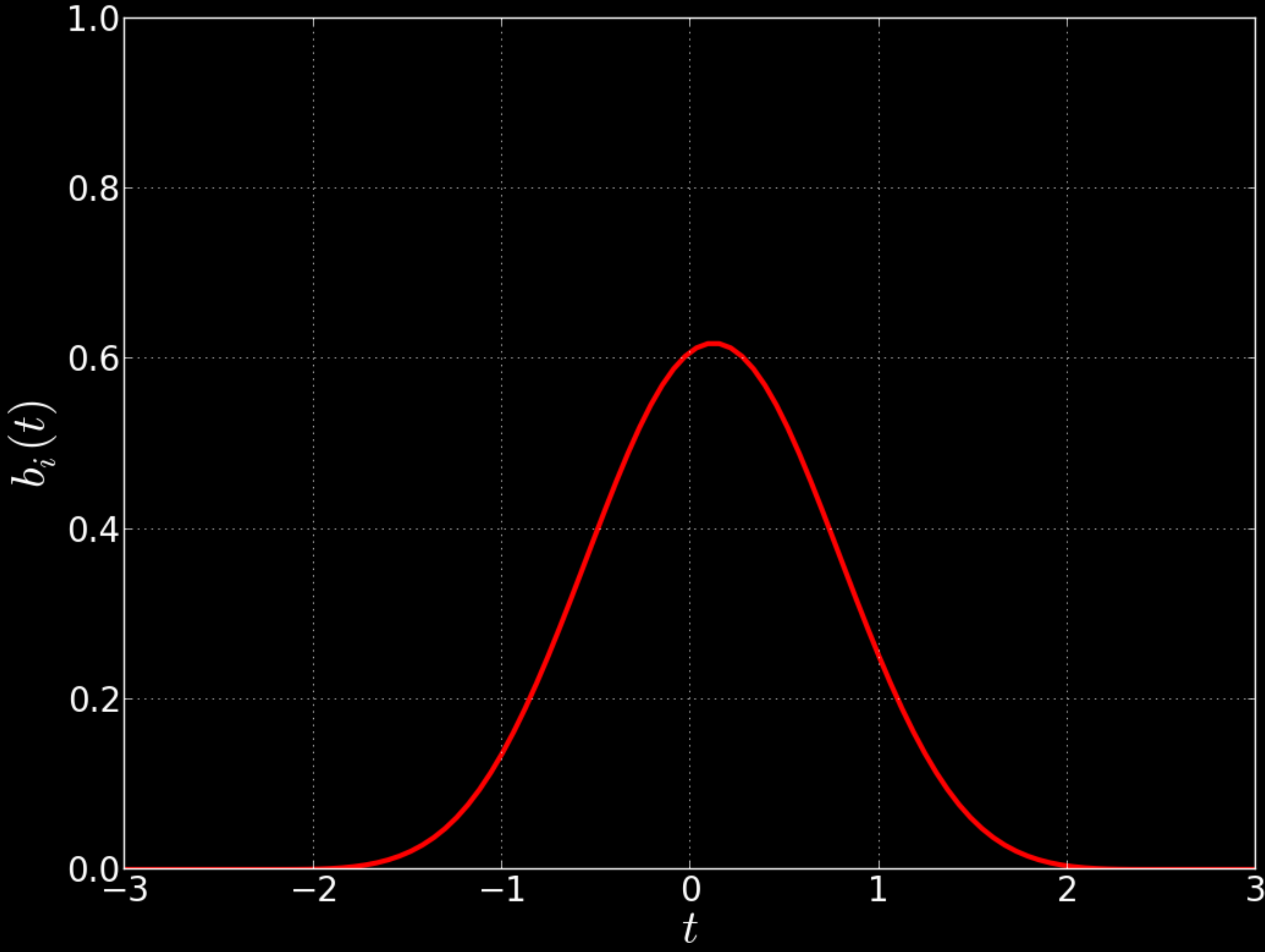


discussion

- global support
- approximation of “long” curves
- geometric continuity

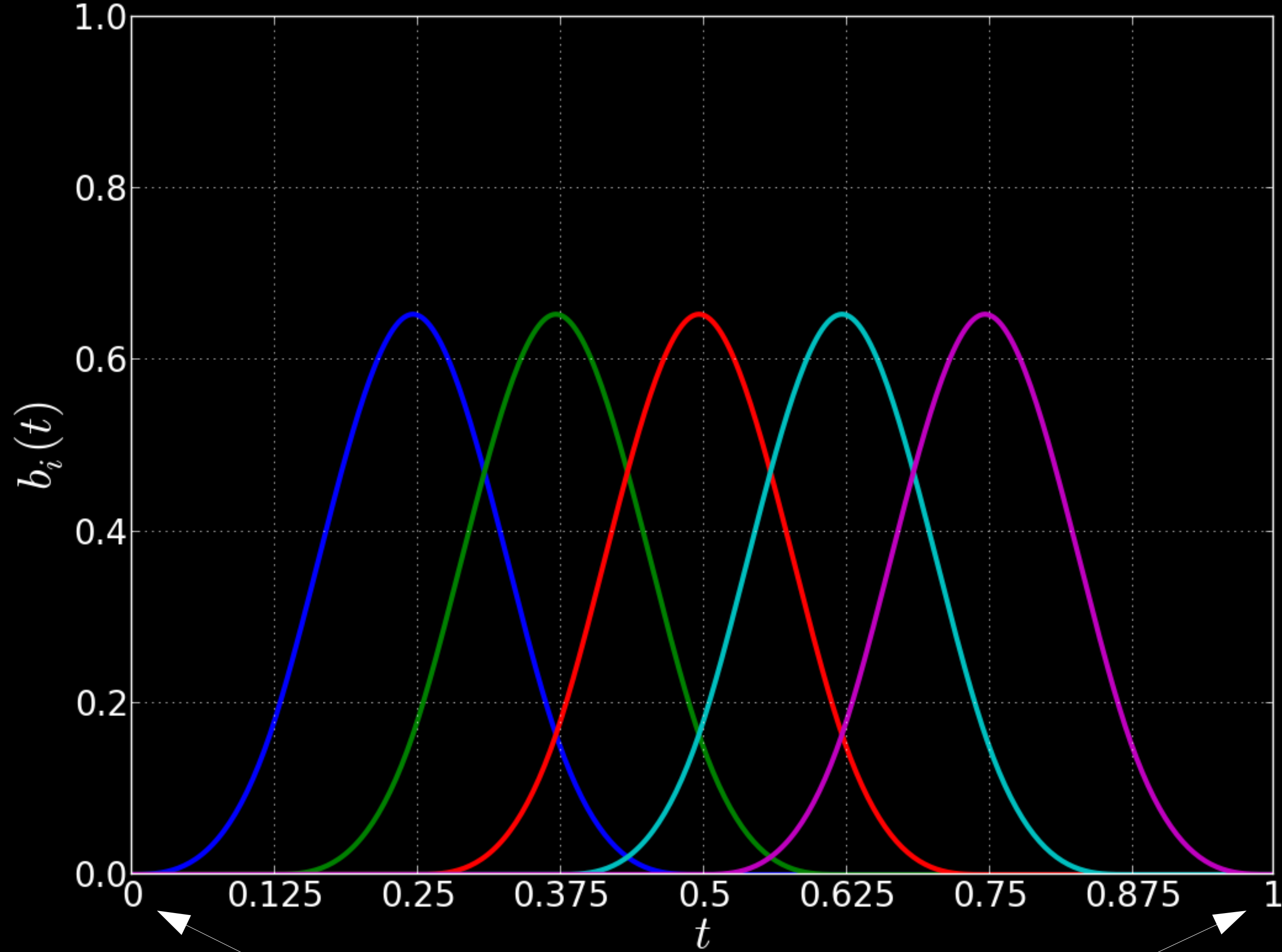
B-splines

construction



$$b^{p+1} = b^p * \chi_{[-0.5,0.5]}$$

the basis



$$\boldsymbol{\xi} = (\xi_1, \ldots, \xi_{n+p+1})$$ *(knot vector)*

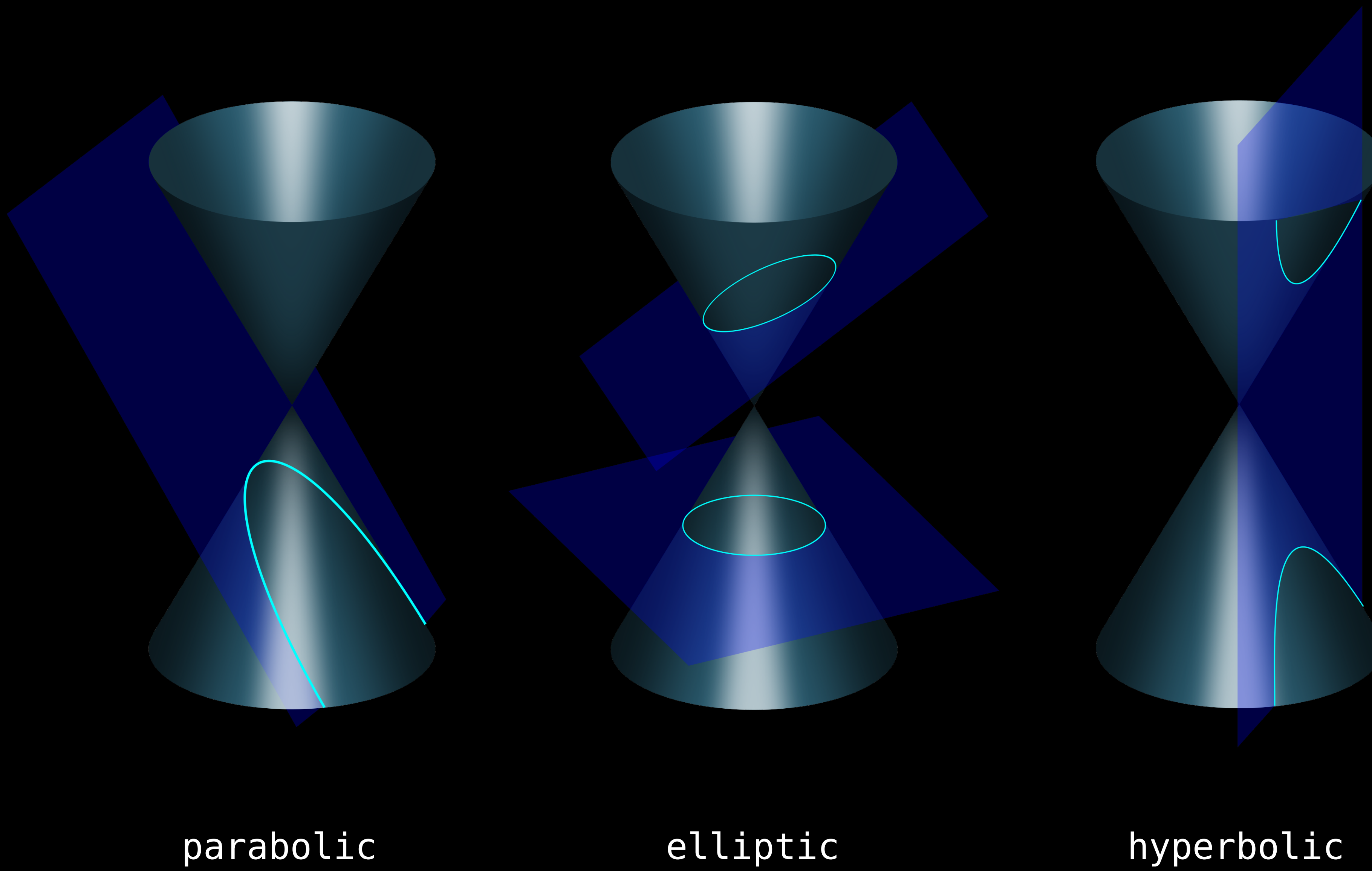
open knot vectors

$$\xi = (0, 0, 0, 0, 0.5, 1, 1, 1, 1)$$

Cox-de Boor algorithm

- recursive
- no convolutions needed
- knots as parameters
- generalization of de Casteljau's algorithm
- fast & numerically stable

conic sections



a glimpse of nurbs

- non-uniform rational b-splines
- control points in $\mathbb{P}_d$
- in homogeneous coordinates:

$$\boldsymbol{c}_i = (x_i, y_i, z_i, w_i)^\top$$

- control points in $\mathbb{R}^d$ weighted by the inverse of w_i

spline surfaces

- two coordinates (u, v)
- linear combination of bi-variate basis functions:

$$\boldsymbol{s}(u, v) = \sum_{k=1}^{m \cdot n} \boldsymbol{c}_k b_k(u, v)$$

- basis by forming

$$b_k(u, v) = b_i(u) b_j(v), \quad i = 1, \ldots, m,\ j = 1, \ldots, n$$

(*tensor product*)

tensor product basis

properties of splines

- linear precision
- convex hull property
- variation-diminishing
- affine invariance

some more blender

applications